





**Towards an Active Schema Integration
Architecture for Heterogeneous
Database Systems**

M.P. Reddy
Michael Siegel
Amar Gupta

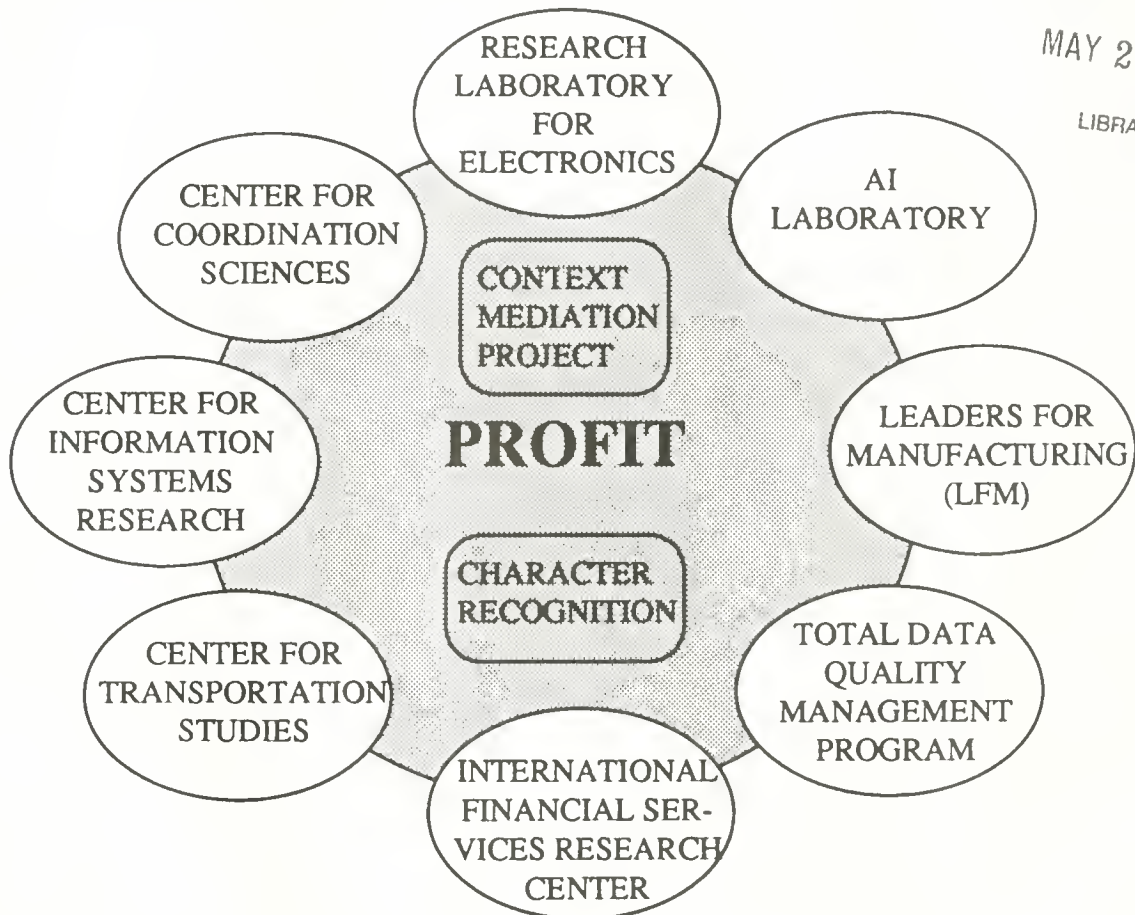
WP #3768 April 1993
PROFIT #93-07

Productivity From Information Technology
"PROFIT" Research Initiative
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
(617)253-8584
Fax: (617)258-7579

Copyright Massachusetts Institute of Technology 1993. The research described herein has been supported (in whole or in part) by the Productivity From Information Technology (PROFIT) Research Initiative at MIT. This copy is for the exclusive use of PROFIT sponsor firms.

Productivity From Information Technology (PROFIT)

The Productivity From Information Technology (PROFIT) Initiative was established on October 23, 1992 by MIT President Charles Vest and Provost Mark Wrighton "to study the use of information technology in both the private and public sectors and to enhance productivity in areas ranging from finance to transportation, and from manufacturing to telecommunications." At the time of its inception, PROFIT took over the Composite Information Systems Laboratory and Handwritten Character Recognition Laboratory. These two laboratories are now involved in research related to context mediation and imaging respectively.



In addition, PROFIT has undertaken joint efforts with a number of research centers, laboratories, and programs at MIT, and the results of these efforts are documented in Discussion Papers published by PROFIT and/or the collaborating MIT entity.

Correspondence can be addressed to:

The "PROFIT" Initiative
Room E53-310, MIT
50 Memorial Drive
Cambridge, MA 02142-1247
Tel: (617) 253-8584
Fax: (617) 258-7579
E-Mail: profit@mit.edu

Towards an Active Schema Integration Architecture for Heterogeneous Database Systems

M. P. Reddy, Michael Siegel, and Amar Gupta
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139

In this paper we describe our research in the development of a four-layered architecture for Heterogeneous Distributed Database Management Systems (HDDDBMS). The architecture includes the local schema, local object schema, global schema, and global view schema. This architecture was developed to support the propagation of local database semantics (e.g., integrity constraints, context) to the global schema and global view. Constraints propagated to the global level can be used to derive new constraints that could not have been recognized by any of the local components. These constraints are important in significantly reducing query processing costs in the HDDDBMS environment by permitting incorporation of techniques similar to semantic query optimization in the single database environment [CFM84,HZ80,Kin81,SSS91]. These techniques are used on the global query to identify candidate databases and reduce the number of required local databases.

So far local, global and view layers are considered to be defined by passive objects (i.e., without methods). As a result, changes to the semantics at the local schema have to be manually propagated to the global level in order to maintain a set of globally consistent integrity constraints. We are currently investigating the use of active objects as components of our four-layer architecture capable of triggering changes in the semantics to maintain a consistent set of global integrity constraints.

In Section 1, we summarize the key components of the four-layer architecture and describe the derivation of global integrity constraints. In Section 2 we describe the role of semantic query processing at the global level and compares this with existing semantic query optimization techniques. Finally, in Section 3 we present our vision for the use of active objects to maintain the consistency of the mapping knowledge and to maintain global integrity constraint consistency.

1 Integration Model

A methodology for designing a HDDDBMS was proposed in [RPR89,Red90]; this methodology used a four-layered schema architecture: local schemata, local object schemata, global schema, and global view schema as shown in Figure 1. Each layer presents an integrated view of the concepts that characterize the layer below.

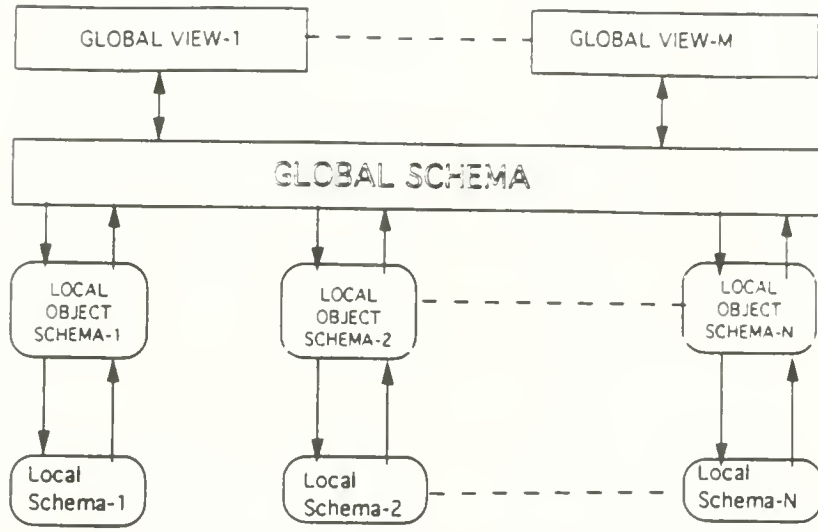


Figure 1: Schema Architecture of a four-layered HDBMS

1.1 Local Schema

The bottom layer consists of a set of local database schemata. Each local database schema is denoted by D_i , where 'i' denotes the identification of the database. These schemata provide the description of the data stored in their respective data models. The stored data can be retrieved only by using their respective query languages.

1.2 Local Object Schema

One local object schema is constructed for each local schema. For a given Local Schema D_i , the construction of its corresponding Object Schema OD_i involves the identification of the set S_i which gives the distinct object types in the schema D_i , the semantic meaning of the data associated with every instance of the object in S_i , and the constraints associated with these objects. The knowledge that maps objects in S_i to their corresponding data structures in D_i is also placed at this layer.

An object in S_i is any distinguishable entity whose description is available in the Local Schema D_i . A database object is denoted by O_l where l is a unique object identifier: l consists of a pair of indices, say (i, j) , where the first index i specifies the schema identification and the second index j provides the object identification within the schema. Each object possesses a set of properties. A property is denoted by P_k where k is a unique property identifier; k is expressed as a pair l, p_i where l is its object identifier and p_i is the property identifier with respect to the object O_l . The Property Set associated with the object O_l is denoted by PS_{O_l} . The object O_l is characterized by its properties. This characterization is denoted by: $O_l \iff PS_{O_l}$. The key property of the object O_l is denoted by $K_l \in PS_{O_l}$.

A property can itself be characterized by a set of meta-properties. Meta-properties are the parameters needed to provide a complete semantic meaning to the symbols associated with the property. For example, PERIODICITY-OF-PAY and CURRENCY represent the meta-properties of the property T-SAL.

Let M^{P_k} denote the set of meta-properties associated with the property P_k and $|M^{P_k}|$ denote the number of meta-properties associated with P_k . For each meta-property there is a set of legal meta-values. DOLLAR, RUPEE, and POUND are some of the meta-values for the meta-property CURRENCY; similarly WEEKLY, MONTHLY, and YEARLY are some of the meta-values for the meta-property PERIODICITY-OF-PAY. Further, if V_k^i is the meta-value of the property P_k associated with the meta-property M_k^i , we define $M_k^i(P_k) = V_k^i$. These meta values are used to recognize semantic incompatibilities among the similar concepts in different layers.

1.3 Global Schema

The global schema is derived from the component local schemata. Objects in the component schemas are first pooled together and then decomposed into object equivalence classes comparing their real world states [NEL86]. Two objects belonging to an equivalence class means they must have the same real world states. Each object equivalence class gives one global object type. Further each local object in an object equivalence class constitutes a component of the global object derived from the object equivalence class. If O_L is a global object and O_l is its component, then we denote this relation as $O_l \in O_L$.

To compute the properties of a global object, we compute the union of the properties of all its components and decompose this union into a number of property equivalence classes where each property equivalence class provides one property for the global object. All properties in one property equivalence class are called components of the global property derived from the particular property equivalence class. If P_L is a global property and P_l is its component, then we denote this relation as $P_l \in P_L$. The semantic meaning for a global property is fixed by defining all the meta-values to the respective meta-properties. Two transformation maps $T_{l,L}$ and $T_{L,l}$ are defined which make P_l semantically compatible to P_L and P_L semantically compatible to P_l respectively.

Two properties P_l and P_L are said to be meta-value compatible with respect to the meta-property M^i if and only if $M^i(P_l) = M^i(P_L)$, that is, if and only if $V_l^i = V_L^i$, and this compatibility is denoted by:

$$P_l \stackrel{M^i}{\sim} P_L$$

If T-SAL is the monthly salary paid in rupees and FAC-P is the annual salary paid in dollars, these two properties are not meta-value compatible with respect to PERIODICITY-OF-PAY or CURRENCY.

• Transformation Map

If a property P_l is not meta-value compatible with P_L with respect to the meta-property M^j , then it is possible to define a transformation map $t_{P_l, P_L}^{M^j}$ which makes P_l meta-value compatible with P_L with respect to the meta-property M^j . Note that $t_{P_l, P_L}^{M^j}$ may be a look-up table.

$$t_{P_l, P_L}^{M^j}(P_l) \stackrel{M^j}{\sim} P_L$$

In the above example, F-PAY is not compatible with T-SAL with respect to the meta-property CURRENCY. The meta-value compatibility can be obtained with the transformation map $t_{T-SAL, F-PAY}^{CURRENCY}$.

As such

$$t_{T-SAL, F-PAY}^{CURRENCY}(T-SAL) \quad CURRENCY \quad F-PAY$$

Here $t_{T-SAL, F-PAY}^{CURRENCY}(T-SAL)$ is $\frac{1}{24}$ times T-SAL, assuming \$1 = Rupees 24.

• Composite Transformation Map

Two properties P_l and \mathbf{P}_L in $[P_k]$ are defined to be semantically compatible with each other if and only if they have meta-value compatibility with respect to all meta-properties pertinent to these properties. This is symbolically denoted by $P_l \sim \mathbf{P}_L$. Further, if P_l and \mathbf{P}_L are not semantically compatible, then the composite transformation map T_{P_l, \mathbf{P}_L} can be defined which makes P_l semantically compatible with \mathbf{P}_L .

Suppose $t_{P_l, \mathbf{P}_L}^1, t_{P_l, \mathbf{P}_L}^2, \dots, t_{P_l, \mathbf{P}_L}^{|M^{P_L}|}$ are the transformation maps which make P_l meta-value compatible with \mathbf{P}_L with respect to the meta properties $M^1, M^2, \dots, M^{|M^{P_L}|}$ respectively. The transformation map can be defined as follows:

$$\begin{aligned} T_{P_l, \mathbf{P}_L}(P_l) &= (t_{P_l, \mathbf{P}_L}^1 \circ t_{P_l, \mathbf{P}_L}^2 \circ \dots \circ t_{P_l, \mathbf{P}_L}^{|M^{P_L}|})(P_l) \\ &= t_{P_l, \mathbf{P}_L}^1(t_{P_l, \mathbf{P}_L}^2(\dots(t_{P_l, \mathbf{P}_L}^{|M^{P_L}|}(P_l)))) \\ &\sim \mathbf{P}_L \end{aligned}$$

Note that if P_l and \mathbf{P}_L are already compatible with respect to a particular meta-property, then the corresponding transformation map can be ignored in the construction of the composite transformation map. By using composite transformation maps, homogeneity among the component properties can be achieved, thereby resolving semantic incompatibilities during the stages of query decomposition and data integration.

This completes the integration of objects having the same real world states.

If the real world state of an object \mathbf{O}_L is contained in that of another object \mathbf{O}_K , then \mathbf{O}_K is said to have super class relationship with \mathbf{O}_L . If there is a superclass subclass relationship among global object types, then the subclass object inherits all the properties of the super class object. All these global objects form the third layer in the architecture.

In the next subsection we discuss a method for deriving the constraints associated with the global schema from the constraints available at the component local object schemata.

1.3.1 Constraints at Global Schema Level

The semantic knowledge of the global schema and the mapping knowledge between the global schema and the component local object schemata are used to transform the constraints on the local object schemata into a set of global constraints. A detailed procedure for this transformation is presented in [RPG92].

Certain constraints are relevant only at the global schema level but not at any of the component local object schema. For example, consider the two relational schemata shown in Figure 2. The

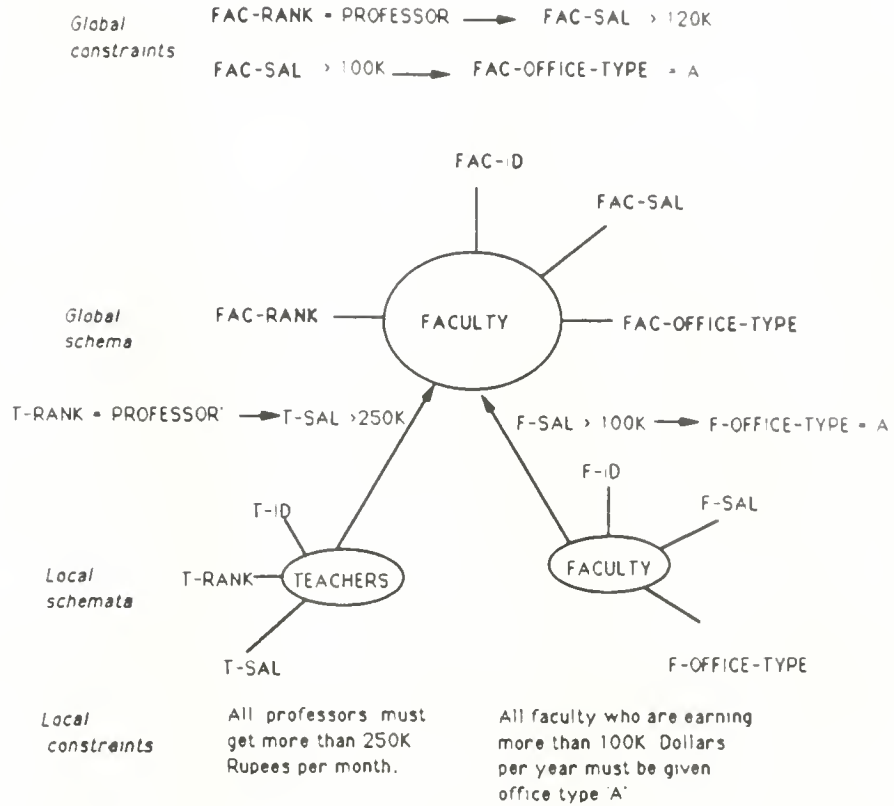


Figure 2: Example Derivation of Global Integrity Constraints

constraints shown at the local level are propagated to the global level [RPG92] and used to derive new global constraints. For example, the constraint $\text{FAC-RANK} = \text{'Professor'} \rightarrow \text{FAC-OFFICE-TYPE} = \text{'A'}$ can only be derived at the global level.

The above discussion shows the meaningful interaction among different layers depends on the semantics of the similar concepts in different layers and in turn depends on the correctness of the composite transformation maps defined between these layers. Our previous work suggest that these composite transformation maps need to be redefined manually whenever the semantics of the concepts present in these layers change.

1.4 View Object Schema

Some of the objects in the third layer may possess disjoint or overlapping domains. The integration of these objects may be required for global users, creating a need for generalizing such objects to produce global views. Each of the global objects that is generalized to produce the global view is called the component of the view object.

The properties of the global view object are derived by first computing the union of the properties of the component objects. This union is decomposed into property equivalence classes; from these we create a subset retaining a property equivalence class only if it contains one property from each and every component of the view object. Each such property equivalence class provides one property for the global object.

The following section outlines the potential benefits of the global integrity constraints and the need to maintain their consistency.

2 Using GICs in Semantic Query Processing

In [RSG92] we describe algorithm for using GICs in semantic query processing. Significant savings can occur using semantic query processing for global queries. Some of the key optimization techniques introduced in our GIC-based query processing strategy are:

- **Null Queries:** Rejection of null global queries at the initial stage would reduce the average query response time. Null queries are typically entered by users who do not possess adequate knowledge about explicit and implicit relations among the objects/entities. This is especially true in a HDDBMS environment where the global schema is generally large and difficult for the user to understand completely.
- **Deduction of Query Results:** SQP facilitate deduction of values of target attributes using available semantic knowledge and query qualification. The deduction of all target attributes may result in answering complete queries. Even when all the target properties may not be deducible using semantic knowledge, the deduction of a subset of the target properties may eliminates the need for the generation of one or more subqueries.
- **Avoidance of Large Search Space:** Because the search space comprises of the union of all the component databases, the time to process global queries may exceed an acceptable range. The need for an exhaustive search of all the component databases can be avoided by implementing a sophisticated query optimization strategy. SQP techniques can reduce the size of the relevant search space by selecting an appropriate minimal set of candidate databases.
- **Optimization of Subqueries:** Semantic query processing does not terminate at the global schema level after optimizing the global query. Subqueries of the global query need to be optimized further by using additional Semantic Query Optimization techniques.
- **Generation of Missing Data:** One of the problems faced during the integration of partial results is that of missing data. This problem arises because of incompleteness of the component databases. This problem may be resolved using GICs.
- **Resolution of Data Inconsistencies:** Data inconsistency is another problem which demands solution during the stage of integration of partial results obtained by processing subqueries against their respective databases. This problem arises because of uncontrolled redundancy inherent in heterogeneous environments. Semantic knowledge can be utilized to overcome this problem.

This semantic query processing concept requires a set of consistent global integrity constraints. However, changes in local database semantics is not easily reflected in the structure or semantic knowledge at the global level. In the following section we provide some insight into how a more active architecture may be able to provide a consistent global representation.

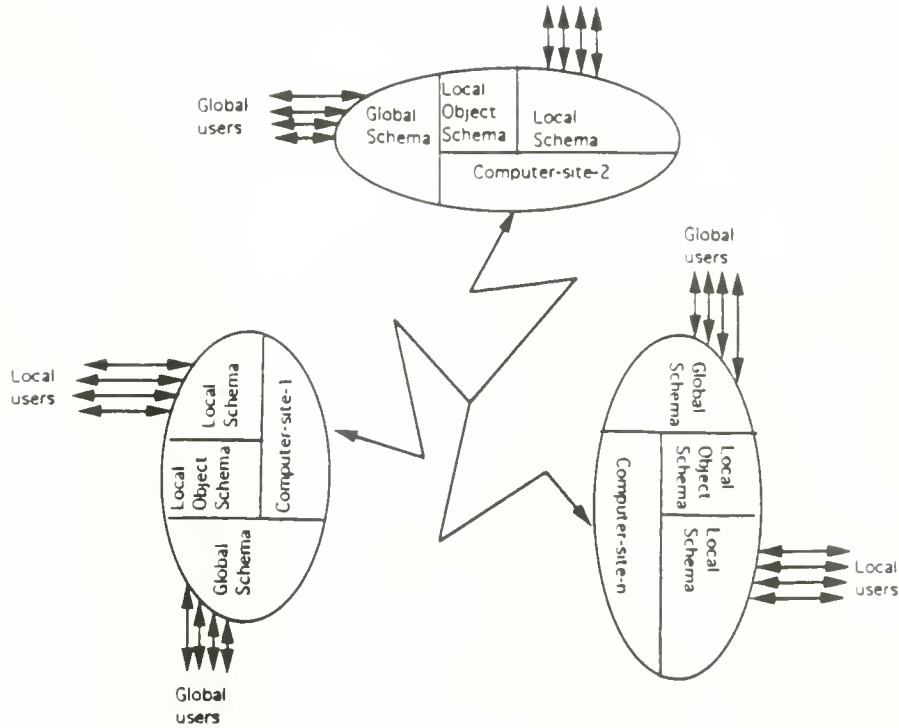


Figure 3: Architecture of the proposed HDDBMS

3 Maintaining Consistent Global Integrity Constraints

Query processing in a HDDMS can be improved using GICs. This is contingent upon the availability of a consistent set of GICs. Since these constraints are derived from the local constraints, any change in the semantics of the local schema impacts the set of local integrity constraints associated with that schema. The corresponding change must be reflected in the GICs. Currently, objects in the local object schema and in the global schema are passive, in the sense that they contain no methods and must be redefined whenever there is any change in the local schema. Our plan is to make these objects active, in the sense that whenever some change occurs in the local schema, the objects in the top three layers evolve to cope with the change at the local schema. For example, consider the system architecture shown in Figure 3. In the “passive world”, a local database administrator would contact the global administrator to register a change in the local schema, and the global administrator would change the local object schema and all other layers and distribute new copies of the global schema to the user sites. In the “active world”, the changes in the local schema would be reflected in the local object schema and inconsistencies would be identified and, when possible, a consistent global schema could be automatically produced.

The schema evolution process has been studied in the context of object oriented databases [BCG*90]. This work mainly concentrates on the schema evolution process (i) changes to the contents of an object class (e.g., changes to an instance variable or method); (ii) changes to relations among the object classes; (iii) addition or removal of object classes from the schema. Because incremental growth is one of the desired features of a HDDBMS, such schema evolutions is applicable to HDDBMS. Automatic schema evolution makes it easier to add a new database to an existing HDDBMS. However, existing evolution mechanisms are not adequate for our requirement. Whenever any change occurs in the semantics of an attribute in the local schema that the change must be reflected in all transformation maps pertinent to that attribute in different layers; further the corresponding LICs and GICs must be modified. We are currently investigating methods that make objects in

different layers active, so that they can be used in our four layered architecture to generate current composite transformation maps, and to generate consistent global integrity constraints.

Some examples of the uses for active objects include the identification of invalid instances of both transformation mappings and global constraints. The layers of the architecture, transformation maps and global constraints can be provided with methods or message passing capabilities that allow for notification of changes in these object states. For example, assume that the constraint in Figure 2 on the local FACULTY relation is changed so that only those faculty members whose salary is more than 150K Dollars per year will get office-type 'A'. This situation requires that one of the previously generated GICs be made invalid and a new GIC must be generated in its place. We proposed to generate GICs and define *demons* to monitor the changes in its component LICs. Whenever there is a change in one of the components these *demons* invoke a method to reconstruct the GIC suitable to the local changes. If the semantics of F-PAY are changed so that it gives annual salaries in Rupees, then the corresponding transformation map is required to be changed. This method for constructing the composite transformation map may access global ontologies or conversion routine libraries. If such automatic construction is not possible, then we would want the system designer to be automatically notified of the impact of these changes.

The four-layered architecture provides a well-defined set of integration stages. We believe that enhancing this network with active capabilities will allow for automatic recognition and resolution of conflicts that resolve from changes in the semantics at the local schema.

References

- [BCG*90] J. Banerjee, H. T. Chou, J. F. Garza, W. Kim, D. Woelk, N. Ballou, and H. J. Kim. Data model issues for object-oriented applications. In S. B. Zdonik and D. Maier, editors, *Readings in Object-Oriented Database Systems*, pages 161–213, Morgan Kaufmann Publishers, Inc, 1990.
- [CFM84] U. Chakravarthy, D. Fishman, and J. Minker. Semantic query optimization in expert systems and database systems. In *Proceedings of the First Intl. Conference on Expert Database Systems*, pages 326–340, 1984.
- [HZ80] M. Hammer and S. Zdonik. Knowledge-based query processing. In *Proceedings 6th VLDB*, pages 137–146, 1980.
- [Kin81] J. King. QUIST : A system for semantic query optimization in relational databases. In *Proceedings 7th VLDB*, pages 510–517, 1981.
- [NEL86] S. B. Navathe, R. Elmasri, and J. Larson. Integrating user views in database design. *Computer*, 19, 1986.
- [Red90] M. P. Reddy. *Heterogeneous Distributed Database Management Systems: Modeling and Managing Heterogeneous Data*. PhD thesis, School of Mathematics & Computer/Information Science, University of Hyderabad, India, 1990.

- [RPG92] M. P. Reddy, B. E. Prasad, and A. Gupta. Formulation global integrity constraints during derivation of global schema. *In Submission to Knowledge and Data Engineering*, 1992.
- [RPR89] M. P. Reddy, B. E. Prasad, and P. G. Reddy. A methodology for resolving semantic incompatibilities and data inconsistencies in integrating heterogeneous databases. In *Proc. Int. Conference on Management of Data, Hyderabad, India*, 1989.
- [RSG92] M. P. Reddy, M. Siegel, and A. Gupta. Semantic query processing in hddbms. *In submission to VLDB Journal*, 1992.
- [SSS91] M. Siegel, S. Salveter, and E. Sciore. Automatic rule derivation for semantic query optimization. *Accepted for publication to Transactions on Database Systems*, 1991.

MIT LIBRARIES



3 9080 00932 7500

3197

Date Due

